Códigos Detetores e Corretores de Erros

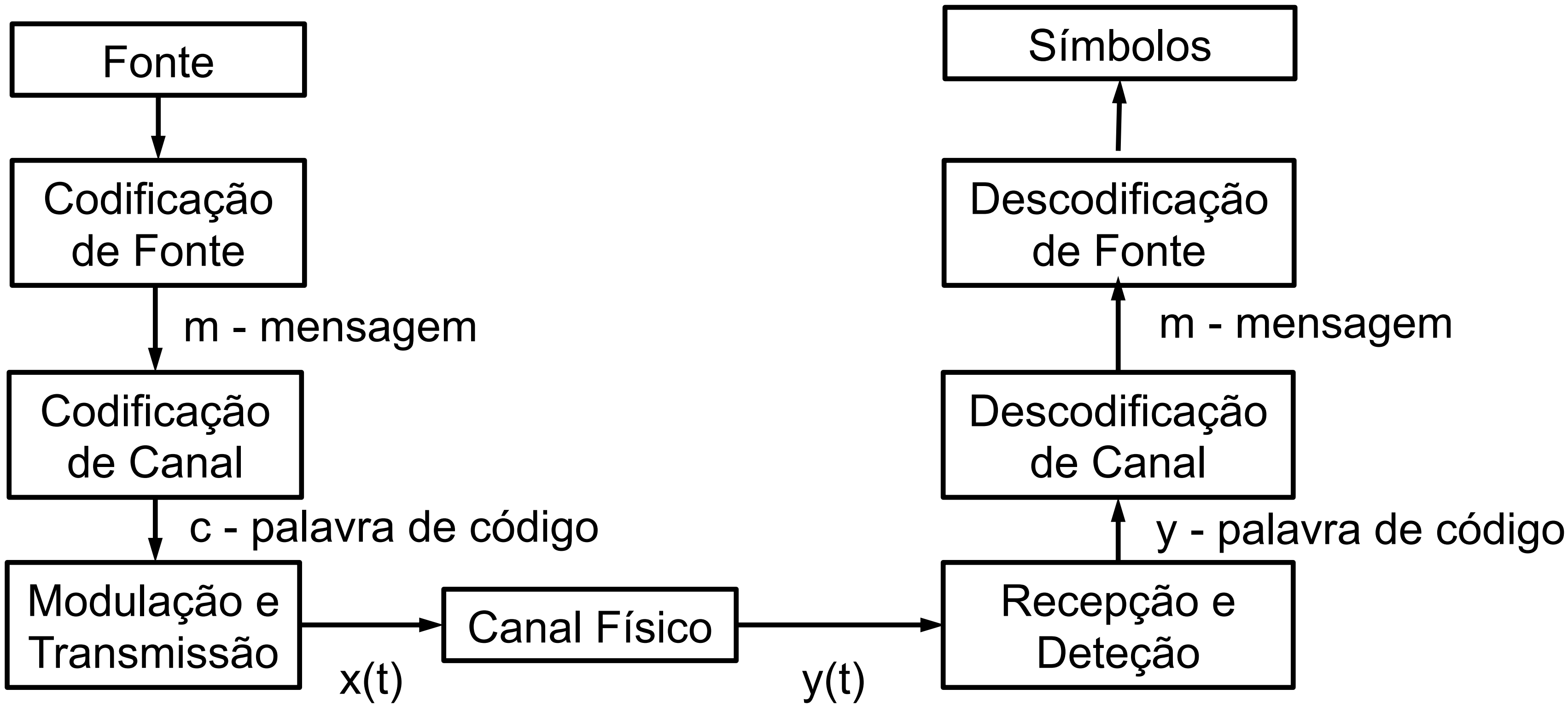
CPS- Comunicação e Processamento de Sinais

Sumário

1. Comunicação Digital e Erros
2. Códigos detetores e corretores de erros
   a) Códigos de bloco linear (n,k)
   b) Características dos códigos
   c) Capacidade de deteção e correção de erros
   d) Códigos de repetição e bit de paridade
   e) Descodificação baseada em síndroma
3. Códigos cíclicos
   a) CRC
4. Aplicações

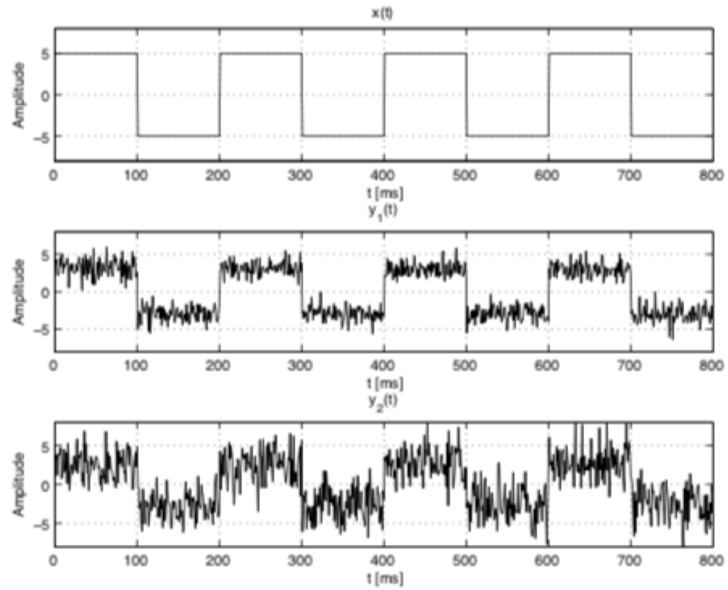
COMUNICAÇÃO DIGITAL E ERROS

Sistema Comunicação Digital

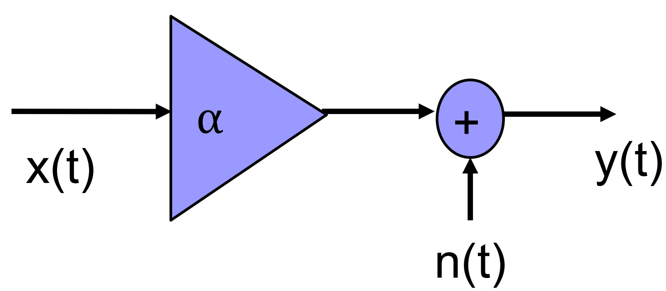


Modelo do Canal Físico

- A correcção e/ou detecção de erros é necessária devido aos erros introduzidos no canal de transmissão ou de armazenamento

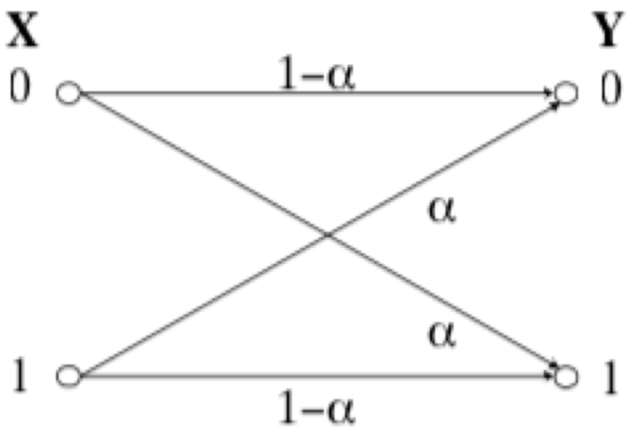


- O modelo AWGN (Aditive White Gaussian Noise) é realista em muitos cenários
  - Ruído aditivo
  - Gaussiano
  - Branco



Modelo de Canal discreto

- O canal é analisado através de um modelo discreto usando variáveis aleatórias discretas
- Modelo Binary Symmetric Channel (BSC)

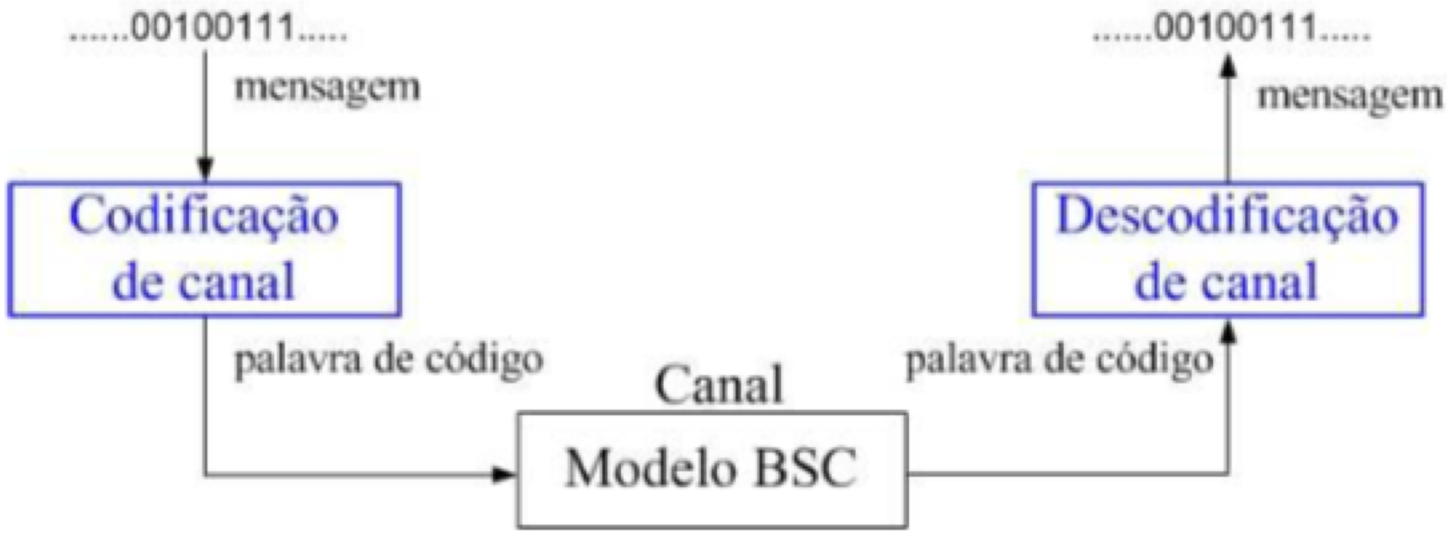


Probabilidade de erro de bit

$$
\begin{aligned}
P_e &= P(y_0, x_1) + P(y_1, x_0) \\
&= P(y_0|x_1)P(x_1) + P(y_1|x_0)P(x_0) \\
&= \alpha P(x_1) + \alpha P(x_0) \\
&= \alpha
\end{aligned}
$$

A probabilidade de Erro α define o Bit Error Rate – BER – do canal, que representa a taxa de erros por bit.

Modelo de Canal Discreto



Implementação computacional modelo BSC:
y = x xor r
r – array com 1s no local onde existe erro

Teorema da Codificação de Canal e Capacidade de Canal

- A probabilidade de erro no canal determina a capacidade C de transferência de informação no canal.

- Teorema da Codificação (2º Teorema de Shannon)

Dada a capacidade C do canal, existe uma técnica de codificação tal que a informação pode ser transmitida no canal a um ritmo R=C, com probabilidade de erro arbitrariamente pequena. Se R>C, não é possível transmitir sem erros.

https://mathshistory.st-andrews.ac.uk/Biographies/Shannon/

Codificação de Canal

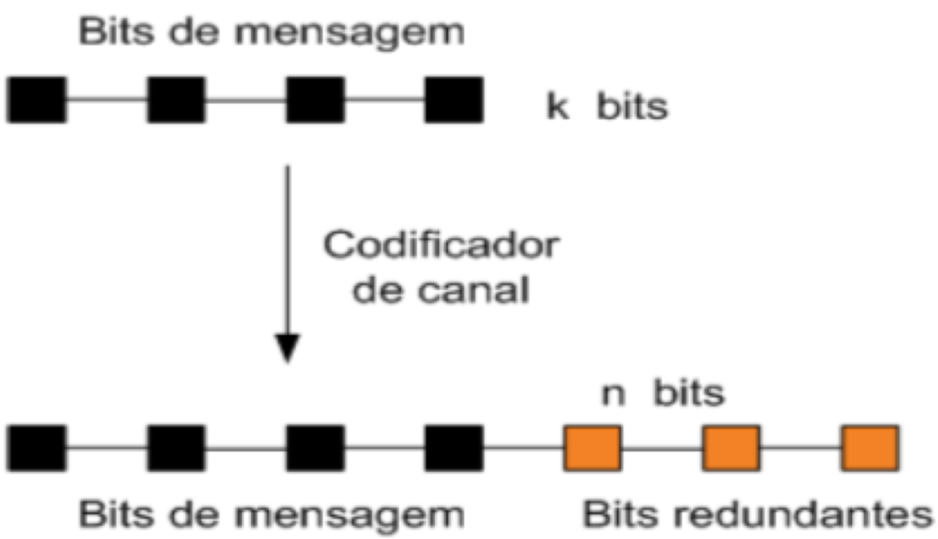
- A deteção e correção consegue-se ao introduzir redundância na mensagem original
- Essa redundância é função da mensagem
- Códigos
  - Códigos de repetição; bit de paridade par; Hamming.
- Os códigos de canal são utilizados nos modos:
  - FEC – Forward Error Correction
  - ARQ – Automatic Repeat reQuest

Modos de Funcionamento

- FEC – Forward Error Correction
  - Modo de Correção de Erros
  - O recetor recebe as palavras de código, deteta eventuais erros e corrige-os
- ARQ – Automatic Repeat reQuest
  - Modo de Deteção de Erros
  - O recetor recebe as palavras de código e deteta eventuais erros; em caso de erro, solicita a retransmissão

Codificação de Canal (n,k)

- Codificador de bloco
- Cada bloco de k bits de mensagem origina uma palavra de código com n bits



Objetivo da Codificação de Canal

- Aumentar a robustez dos sistemas de comunicação

Códigos de Bloco (n,k) Propriedades

- *(n,k): n* – nº bits da mensagem; *k* – nº de bits da mensagem
- Code Rate (ritmo) $R = \frac{k}{n}$, medida de eficiência
- Distância de Hamming (dH): número de dígitos em que diferem duas quaisquer palavras do código
- Distância mínima (dmin): é a menor distância de Hamming entre duas quaisquer palavras de código; depende da redundância introduzida

$$d_{min} \leq 1 + q, \;\; q = n - k$$

- Detecta todos os padrões até *"l"* erros: $l \leq d_{min} - 1$
- Corrige todos os padrões de erro até *"t"* erros: $t \leq \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$
- Detecta "l" erros e corrige "t" erros: $d_{min} \geq l + t + 1, \;\; l > t$

Códigos lineares de bloco (n,k)

- Bloco: todas as palavras têm a mesma dimensão
- Linear: o vector nulo pertence ao código; a soma modular de duas palavras do código é ainda uma palavra de código

n = número de bits da palavra de código

k = número de bits da mensagem

q = n – k, é o número de bits redundantes

2^n palavras possíveis

2^k palavras de código

Seja $\mathbf{m} = [m_0\ m_1\ \ldots\ m_{k-1}]$ a mensagem e **c** a palavra de código

Podem ser sistemáticos ou não sistemáticos; exemplos destas formas:

- sistemática: $\mathbf{c} = [m_0\ m_1\ \ldots\ m_{k-1}\ \ b_0\ b_1\ \ldots\ b_{q-1}]$
- não sistemática: $\mathbf{c} = [m_0\ b_1\ b_0\ m_1\ \ldots\ m_{k-1}\ \ldots\ b_{q-1}]$

Peso de Hamming

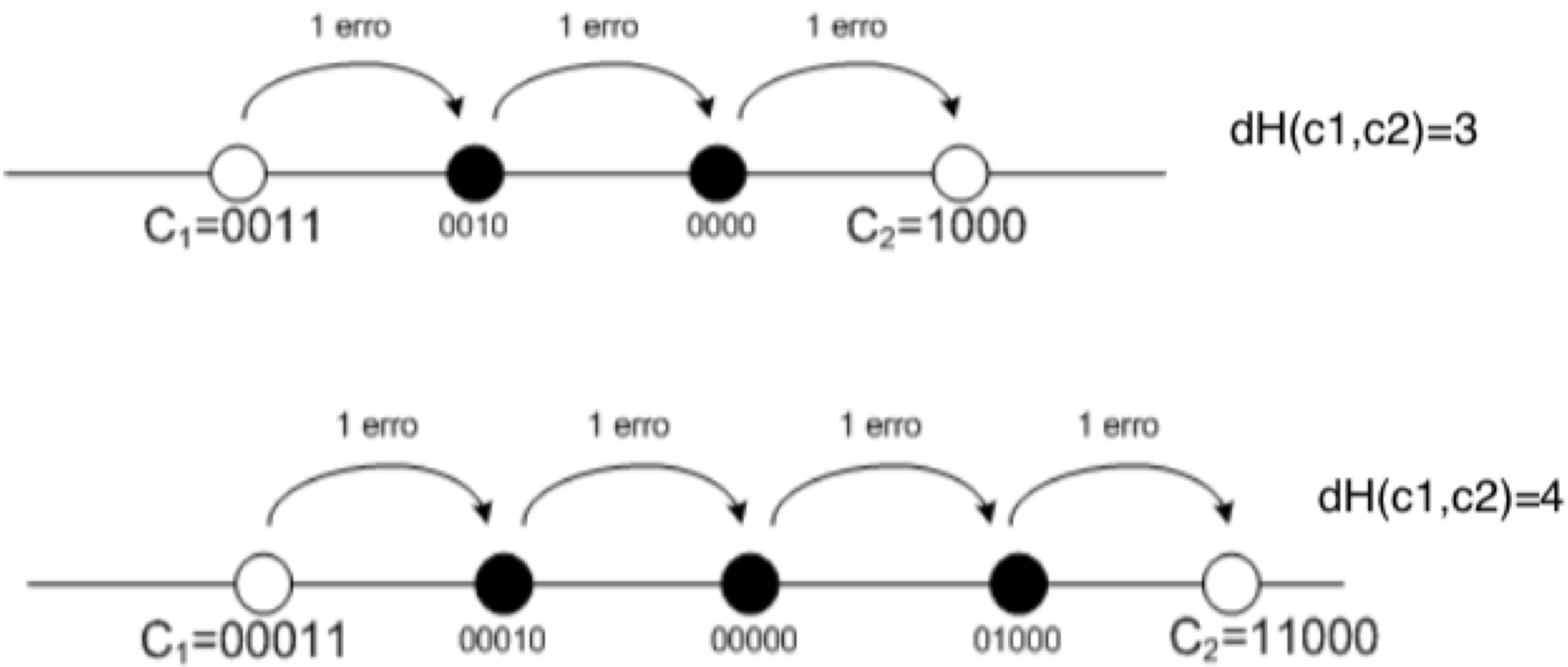
- Define-se peso de Hamming (w) como o número de dígitos não nulos numa palavra
- Sejam c_i e c_j duas palavras distintas de um código linear de bloco; tem-se por definição que $dmin = \min_{i \neq j} dH(c_i, c_j)$
- Dado que o código é linear, tem-se:

$$dmin = \min w(c_i \oplus c_j) = \min w(c_k)$$

sendo c_k palavra do código, diferente do vector nulo

Códigos de bloco (n,k) Distância

- Distância de Hamming entre palavras



Códigos de bloco (n,k)

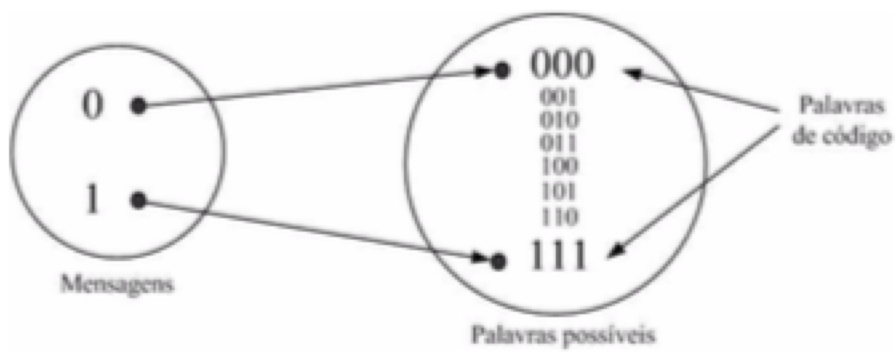
Capacidade de deteção, correção e deteção e correção simultâneas, em função da distância mínima, para uma gama de valores

d_{min}	**Detecção (l)**	**Correcção (t)**	**Det. e Corr. simultâneas (l,t)**
1	0	0	Não tem
2	1	0	Não tem
3	2	1	Não tem
4	3	1	(2,1)
5	4	2	(3,1)

Código de Repetição (3,1)

- Consiste na repetição da mensagem

Mensagem	Palavra de Código
0	000
1	111



Código de Repetição (3,1)

- Descodificação realizada por maioria
- A distância entre as palavras do código garante dmin=3

Exemplos:

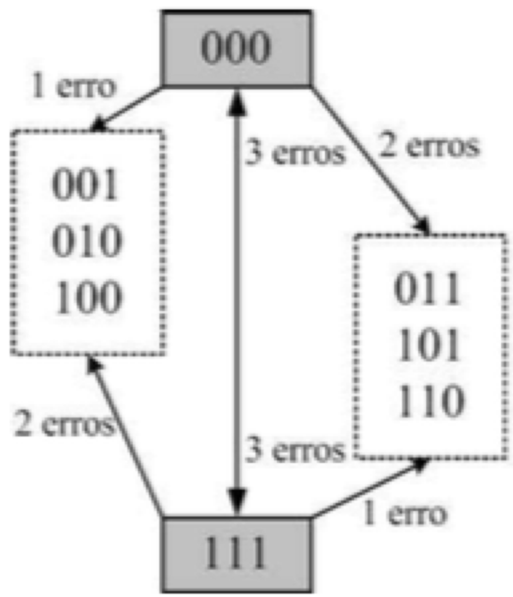
Código de repetição (3,1)

m	c	w(c)
0	000	0
1	111	3

$d_{min} = 3$

$l = 2$

$t = 1$



Código de Repetição (3,1)

Distribuição Binomial

$$f(k;n,p) = \binom{n}{k} p^k (1-p)^{n-k}$$

para $k = 0, 1, 2, \ldots, n$ e onde $\binom{n}{k}$ é uma combinação.

Colocando a função completa, incluindo a Combinação:

$$f(k;n,p) = \frac{n!}{k!(n-k)!}\, p^k (1-p)^{n-k}$$

Cada parte da função acima traduz os seguintes dados:

A combinação $\frac{n!}{k!(n-k)!}$ contém as ordenações possíveis;

O número de sucesso é p^k, e;

A probabilidade de fracassos é $(1-p)^{n-k}$.

Código de Repetição (3,1)

Considerando que se utiliza este código sobre um BSC com $P_e = \alpha = 10^{-5}$ tem-se que a probabilidade de errar 1 bit, sobre uma palavra de 3 bits, é dada por

$$P(1,3) = C_1^3\ \alpha^1(1-\alpha)^2 = \frac{3!}{2!1!}\ \alpha(1-\alpha)^2 = 3\alpha - 6\alpha^2 + 3\alpha^3 \approx 3 \times 10^{-5}, \tag{7}$$

em que C_1^3 representa combinações de três um a um.

A probabilidade de errar 2 bits é

$$P(2,3) = C_2^3\ \alpha^2(1-\alpha)^1 = \frac{3!}{1!2!}\ \alpha^2(1-\alpha) = 3\alpha^2 - 3\alpha^3 \approx 3 \times 10^{-10},$$

Finalmente, a situação extrema de errar os 3 bits da palavra ocorre com probabilidade

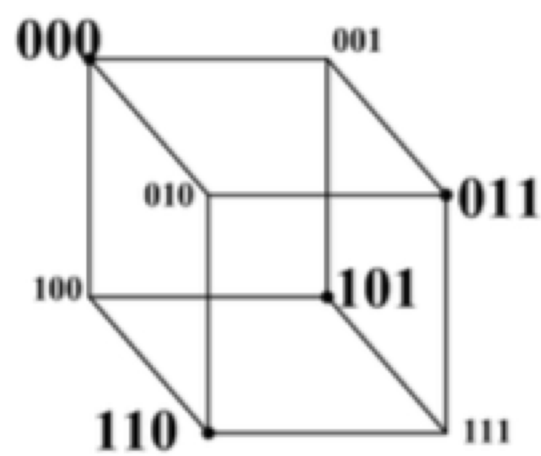
$$P(3,3) = C_3^3\ \alpha^3(1-\alpha)^0 = \frac{3!}{0!3!}\ \alpha^3 = \alpha^3 = 10^{-15},$$

concluindo-se que $P(3,3) \ll P(2,3) \ll P(1,3)$. Verifica-se que a capacidade de correcção até 1 bit errado é adequada nesta situação.

Código de Bit Paridade (3,2) Paridade Par

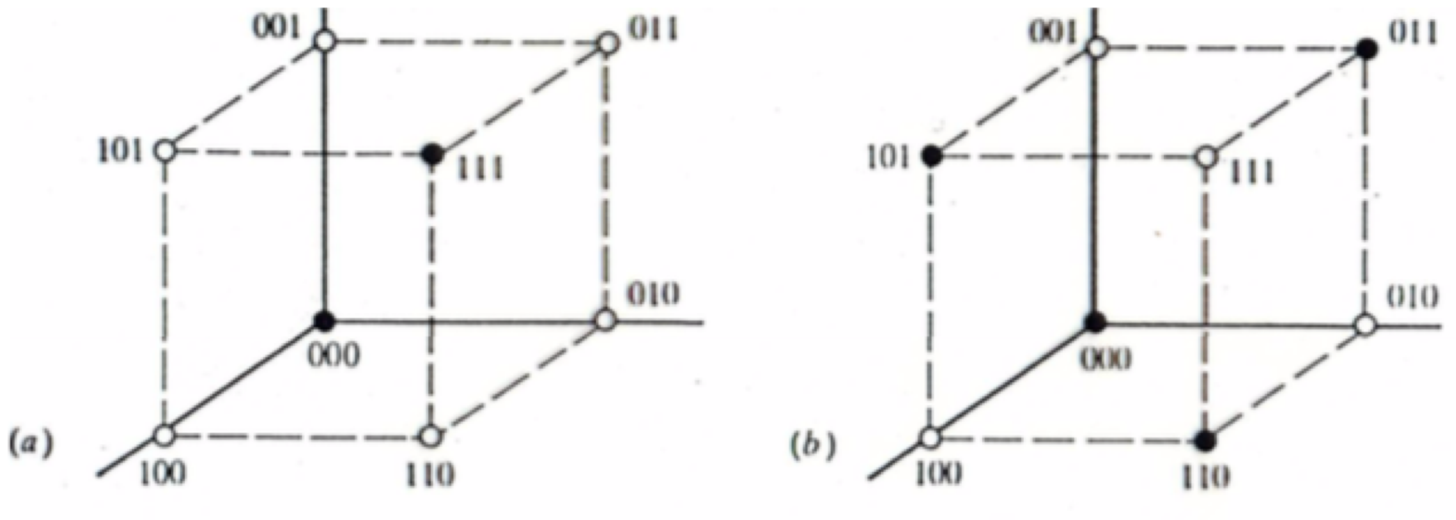
- Adicionar um bit no final da mensagem; este bit é a soma módulo 2 dos bits da mensagem
- A palavra de código $c = [m_0 \quad m_1 \quad m_0 \oplus m_1]$

m	c
00	000
01	011
10	101
11	110



- *dmin = 2, l=1, t=0*

Palavras de código como vetores



- Palavra de 3 bit
  (a) Código de repetição (3,1): 3 arestas entre as 2 palavras de Código
  (b) Código de bit de paridade (3,2): 2 arestas entre 2 palavras de Código mais próximas

Matriz Gerador

- A palavra de código c pode ser obtida através do vetor mensagem m pela matriz geradora G:

$$c = m \times G$$

- c é o vetor de dimensão 1x n; m é o vetor 1x k
- G é a matriz k x n; nos códigos sistemáticos tem-se:
  - $\mathbf{G} = [\ \mathbf{I_k} \mid \mathbf{P}\ ]$ ou $\mathbf{G} = [\ \mathbf{P} \mid \mathbf{I_k}\ ]$ sendo **P** a **sub-matriz geradora de paridade**, ou seja, a matriz que estabelece as equações de paridade do código
  - Cada coluna de **P** constitui uma equação de paridade
  - **P** tem dimensões k x q

Matriz Geradora - exemplos

- Exemplos de matrizes geradoras
- Código de repetição (3,1)

$$G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

- Código de bit de paridade par (3,2)

$$G = \begin{bmatrix} I_2 \, P \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{ou} \quad G = \begin{bmatrix} P \, I_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Códigos de Hamming

- Família de códigos lineares de bloco
- Têm dmin = 3, logo corrigem todos os erros de 1 bit
- Definidos por um parâmetro inteiro m (≥ 2) tal que:

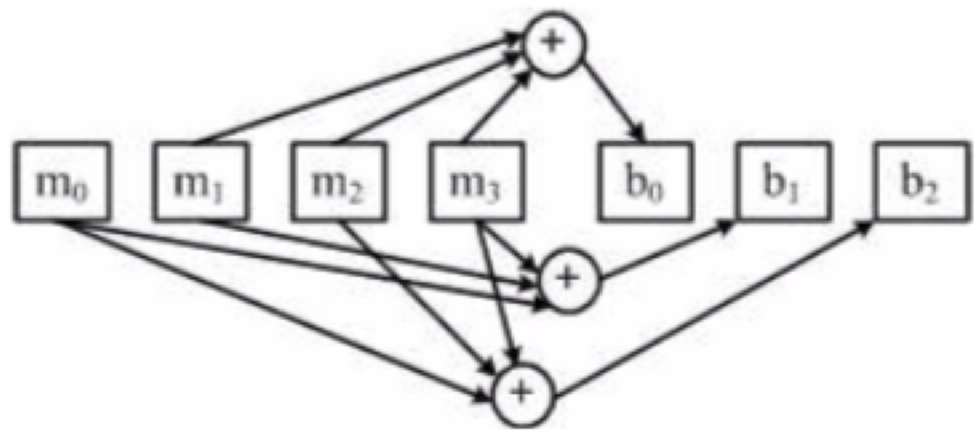
$$(\mathrm{n},\mathrm{k}) = (2^m - 1, 2^m - 1 - \mathrm{m})$$

r	(n,k)
2	(3,1)
3	(7,4)
4	(15,11)
5	(31,26)
6	(63,57)
...	...

- Exemplo m=3, tem-se o código (7,4):

$$\mathbf{c} = [m_0\ m_1\ m_2\ m_3\ b_0\ b_1\ b_2].$$

$$b_0 = m_1 \oplus m_2 \oplus m_3,$$
$$b_1 = m_0 \oplus m_1 \oplus m_3,$$
$$b_2 = m_0 \oplus m_2 \oplus m_3,$$



Códigos de Hamming
Forma Matricial – Hamming (7,4)

$$
\begin{aligned}
\mathbf{c} &= m\mathbf{G} = m\,[I_4 \mid P] \\
&= \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix} \\
&= \begin{bmatrix} m_0 & m_1 & m_2 & m_3 & b_0 & b_1 & b_2 \end{bmatrix}
\end{aligned}
$$

Exemplo palavras de código geradas

Palavra de código							Peso
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	4
0	0	1	0	1	0	1	3
0	0	1	1	0	1	0	3
0	1	0	0	1	1	0	3
0	1	0	1	0	0	1	3
0	1	1	0	0	1	1	4
0	1	1	1	1	0	0	4

Palavra de código							Peso
1	0	0	0	0	1	1	3
1	0	0	1	1	0	0	3
1	0	1	0	1	1	0	4
1	0	1	1	0	0	1	4
1	1	0	0	1	0	1	4
1	1	0	1	0	1	0	4
1	1	1	0	0	0	0	3
1	1	1	1	1	1	1	7

O menor peso de Hamming para as palavras não nulas é 3, logo:

$$\mathrm{dmin} = 3,\ l = 2 \text{ e } t = 1$$

Códigos de Hamming Características

- Seja k o número de bits da mensagem e n o númeor de bits efetivamente transmitidos
- Códigos de Hamming são códigos de bloco linear (n,k):
  - $q \geq 3$, sendo $q = n - k$ o número de bits redundantes
  - $n = 2^q - 1$
  - para $q = \{1,2,3,\cdots\}$, temos então (7,4), (15,11), (31,26), …
  - A eficiência do código é $r = k/n = 1\text{-}q\,/\,(2^q - 1)$
    - $r \rightarrow 1$, se $q \gg 1$
  - dmin = 3, independente de q
  - A submatriz P geradora de paridade consiste em k linhas com q bits, com dois ou mais bits a 1 por linha

Descodificador de Canal Caraterísticas

- Recebe a palavra y (possivelmente com erros)
- Estima a palavra de código $\hat{x}$ que lhe deu origem
- Estima a mensagem $\hat{m}$

$$\hat{x} = \arg min_{x_i} dH(y, x_i), \forall x_i \in C$$

- Funciona num dos seguintes modos:
  - Deteção
  - Correção
  - Deteção e correção

Conclusão:
Se a palavra recebida y não pertencer ao código houve erro

Codificação/ Descodificação Forma matricial

- O codificador gera as palavras de código através da matriz geradora G, com
$$c = mG$$
- No caso dos códigos sistemáticos temos $G = [I_k \quad P]$
  - P é a sub-matriz geradora de paridade
- O descodificador verifica se existem erros na palavra recebida c, através do cálculo do síndroma
$$s = cH^T$$
- No caso dos códigos sistemáticos $H = [P^T \quad I_{n-k}]$
- Caso s seja nulo, não se detectam erros
- Caso contrário, existem erros detetados

Codificação/ Descodificação Forma matricial

- Na codificação temos

$$c = mG = m[I_k \quad P] = [m_0 m_1 \dots m_{k-1} b_0 b_1 \dots b_{q-1}]$$

- Na descodificação é necessário obter os bits de mensagem, recalcular a paridade e comparar com os bits de paridade recebidos

$$s = cH^T = mGH^T = m[I_k \quad P]\begin{bmatrix} P \\ I_q \end{bmatrix} = [s_0 s_1 \dots s_{q-1}]$$

Descodificação

- O síndroma é um vetor de q bits (H^Ttem dimensão nxq)
- Cada bit do síndroma corresponde à verificação da presença de erros no respetivo bit de paridade
- Na ausência de erros temos síndroma nulo porqu GH^T são ortogonais

$$s = cH^T = mGH^T = m[I_k \quad P]\begin{bmatrix} P \\ I_q \end{bmatrix} = [00 \ldots 0]$$

- Erros são detetados sempre que o síndroma é não nulo
- O sindroma só depende do padrão de erro e:

$$s = (c+e)H^T = cH^T + eH^T = [00 \ldots 0] + eH^T = eH^T$$

Descodificação

- Cada padrão de 1 bit em erro, tem um síndroma único associado
  - e1 = [1 0 … 0], corresponde ao primeiro bit errado
  - e2 = [0 1 … 0], corresponde ao segundo bit errado
- Para cada palavra de código c (com erro) temos:
  - $s_1 = e_1 H^T$ = primeira linha de H^T
  - $s_2 = e_2 H^T$ = segunda linha de H^T

Nota: as linhas de H^T são sempre não nulas.

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

Table 13.2-1 Codewords for the (7, 4) Hamming code

M	*C*	*w*(*X*)	*M*	*C*	*w*(*X*)
0 0 0 0	0 0 0	0	1 0 0 0	1 0 1	3
0 0 0 1	0 1 1	3	1 0 0 1	1 1 0	4
0 0 1 0	1 1 0	3	1 0 1 0	0 1 1	4
0 0 1 1	1 0 1	4	1 0 1 1	0 0 0	3
0 1 0 0	1 1 1	4	1 1 0 0	0 1 0	3
0 1 0 1	1 0 0	3	1 1 0 1	0 0 1	4
0 1 1 0	0 0 1	3	1 1 1 0	1 0 0	4
0 1 1 1	0 1 0	4	1 1 1 1	1 1 1	7

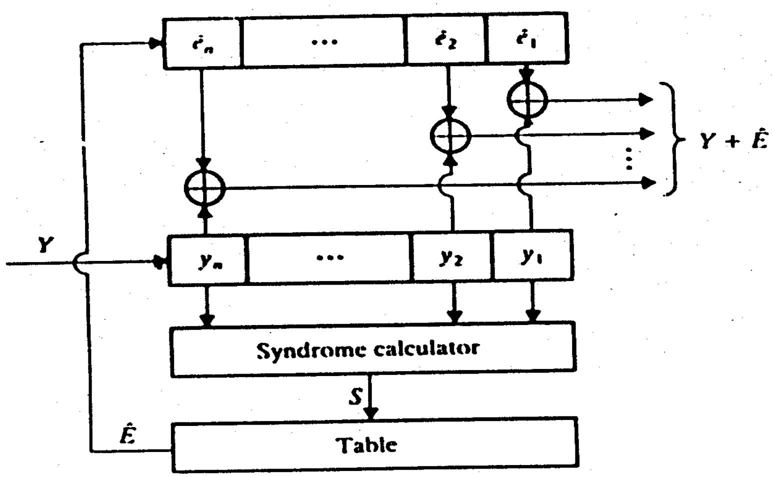


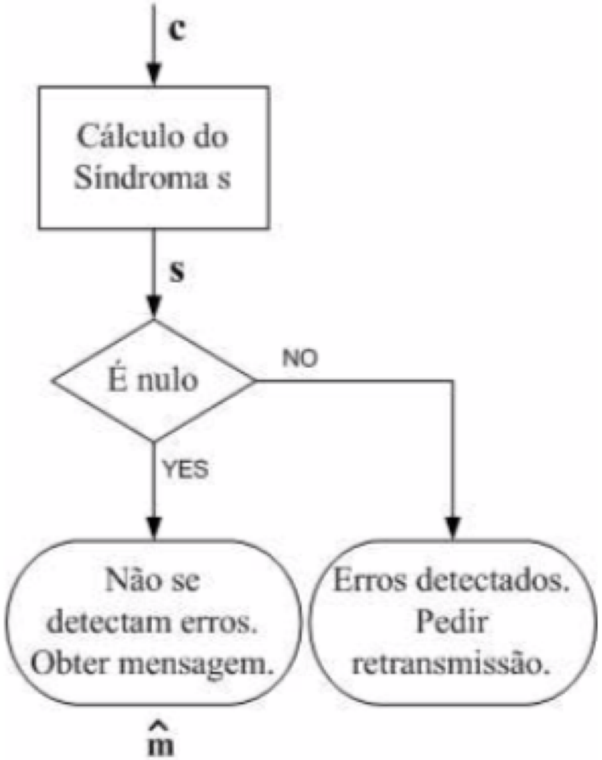
Figure 13.2-2 Table-lookup decoder.

Table 13.2-2 Syndromes for the (7, 4) Hamming code

S	$\hat{E}$
0 0 0	0 0 0 0 0 0 0
1 0 1	1 0 0 0 0 0 0
1 1 1	0 1 0 0 0 0 0
1 1 0	0 0 1 0 0 0 0
0 1 1	0 0 0 1 0 0 0
1 0 0	0 0 0 0 1 0 0
0 1 0	0 0 0 0 0 1 0
0 0 1	0 0 0 0 0 0 1

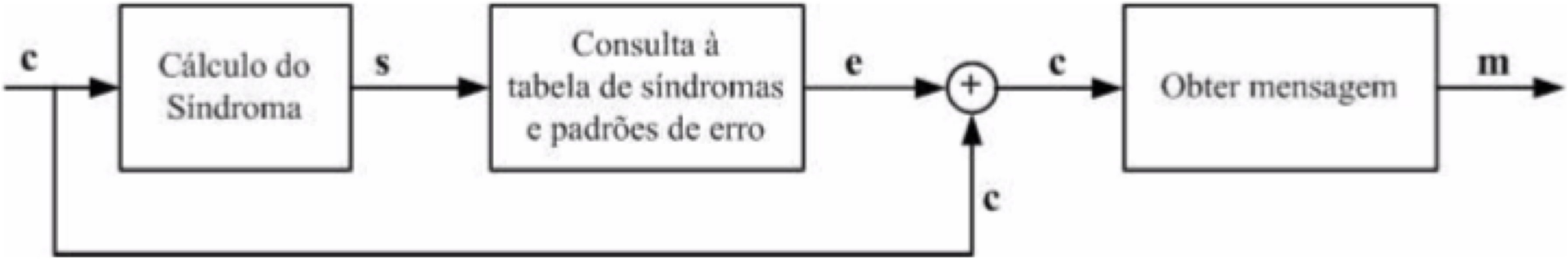
Descodificação: Modo Deteção (ARQ)

- Processo de descodificação em modo de deteção (ARQ)



Descodificação: Modo Correção (FEC)

- Processo de descodificação em modo de correção (FEC)



Comparação de códigos

Código	R = k/n	dmin	Detecta l	Corrige t
Repetição (2,1)	0.500	2	1	0
Repetição (3,1)	0.333	3	2	1
Repetição (4,1)	0.250	4	3	1
Repetição (5,1)	0.200	5	4	2
Paridade (3,2)	0.666	2	1	0
Paridade (8,7)	0.875	2	1	0
Hamming (7,4) m=3	0.571	3	2	1
Hamming (15,11) m=4	0.733	3	2	1
Hamming (31,26) m=5	0.838	3	2	1

Aplicações

- Comunicação série assíncrona
  - 1 bit de paridade por cada byte
- Memórias RAM
  - 1 bit de paridade por cada byte
  - Mais do que 1 bit de paridade – ECC (error correcting code) RAM
- Teletexto
  - Hamming (8,4) – extensão do Hamming (7,4)
- Discos rígidos
  - Alguns usam código de Hamming – existem bits de paridade por cda sector

Aplicações

- RAID (Redundant array of Independent Disks)
  - RAID 1 – mirroring – código de repetição
  - RAID 2 – Hamming – no caso do Hamming (7,4) usa 7 discos (4 de dados + 3 paridade)
  - RAID 3 – parallel transfer with parity drive; usa código bit paridade, existem vários discos de dados e um de paridade
- Bluetooth (comunicação sem fios)
  - Usa código de repetição (3,1) - packet header
  - Usa Hamming modificado (15,10) – application data

Códigos Cíclicos

- Nos códigos cíclicos tem-se que qualquer rotação cíclica de qualquer ordem sobre uma palavra de código é ainda uma palavra de código

- Exemplo: código de bit de paridade par (3,2)

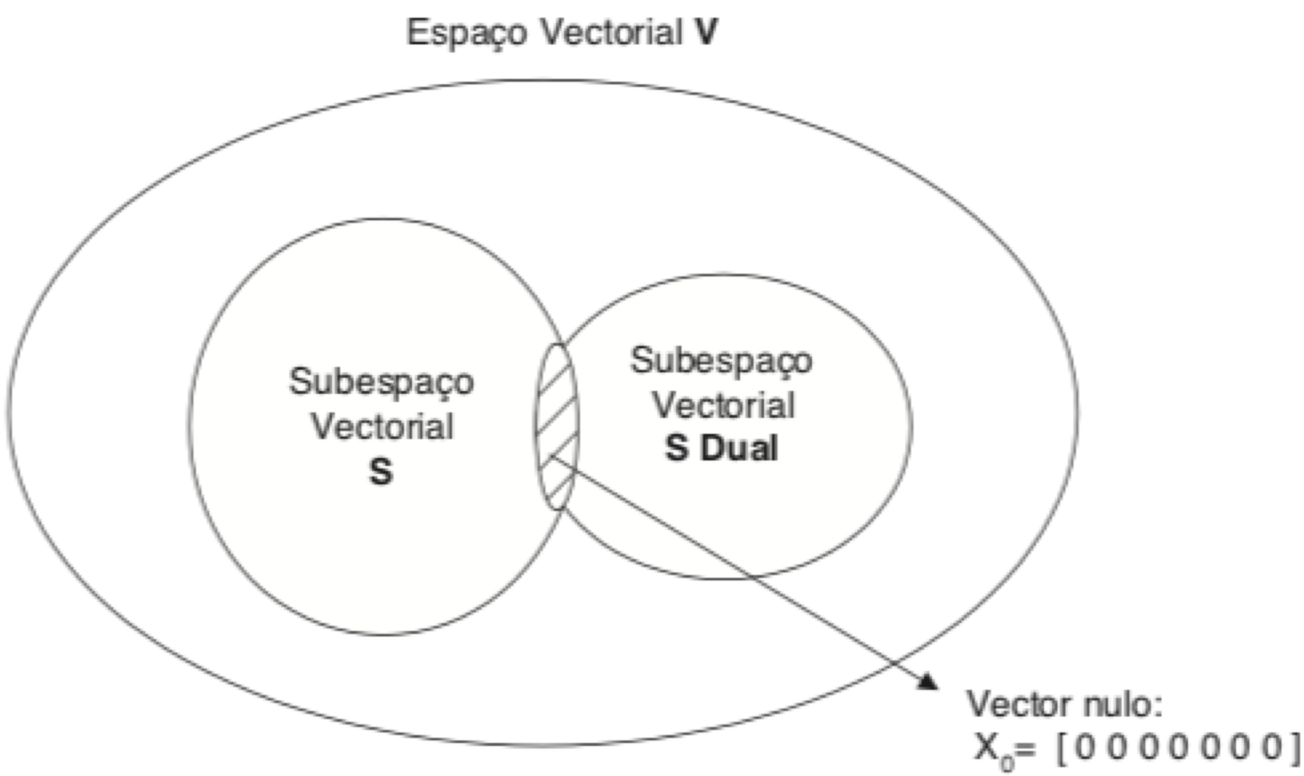
m	c
00	000
01	011
10	101
11	110

- Rotações sobre a palavra de código $\mathbf{c} = [c_{n-1}\ c_{n-2}\ \ldots\ c_1\ \ c_0].$

$$\mathbf{c}' = [c_{n-2}\ c_{n-3} \ldots\ c_0\ c_{n-1}]. \qquad \mathbf{c}'' = [c_{n-3}\ \ldots\ c_0\ c_{n-1}\ c_{n-2}].$$

Código lineares de bloco cíclicos

- Códigos lineares de bloco e cíclicos



Palavras de código como polinómios

- A palavra de código $c = [c_{n-1}c_{n-2} \ldots c_1c_0]$ podem ser analisadas como polinómios:

$$c(X) = \ c_{n-1}X^{n-1} + c_{n-2}X^{n-2} \pm \cdots + c_1X + c_0$$

- Tem-se $c(X) = m(X)g(X)$ em que:
  - $c(X)$ é a palavra de código – polinómio de grau n-1
  - $m(X)$ depende da mensagem – polinómio de grau k-1
  - $g(X)$ polinómio gerador de grau q

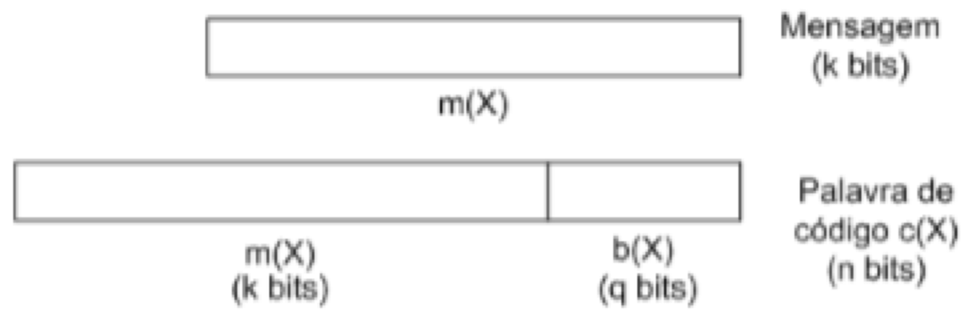
- O número de bits redundates (de paridade) corresponde ao grau do polinómio gerador

Polinómios geradores

Código	Polinómio gerador g(X)
CRC4	$X^4+X^3+X^2+X+1$
CRC7	$X^7+X^6+X^4+1$
CRC12	$X^{12}+X^{11}+X^3+X^2+X+1$
CRC16	$X^{16}+X^{15}+X^2+1$
CRC-CCITT	$X^{16}+X^{12}+X^5+1$
CRC32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

CRC – Cyclic Redundancy Check

- Num código cíclico sistemático, as palavras têm a seguinte organização



- Os bits b(X) que consituem um polinómio de grau q-1 designam-se por CRC – Cyclic Redundancy Check
- A palavra de código é gerada por:

$$c(X) = m(X)X^q + b(X) = m(X)X^q + \text{resto}\left[\frac{m(X)X^q}{g(X)}\right]$$

CRC

- O CRC resulta do resto da divisão de polinómios entre:
  - A mensagem deslocada de que bits para a esquerda
  - O polinómio gerador do código

$$CRC = b(X) = \text{resto}\left[\frac{m(X)X^{q}}{g(X)}\right]$$

Exemplo

- Exemplo de cálculo do CRC para código (7,4)
  - $m(X)=X^3+1=[1\ 0\ 0\ 1]$
  - $g(X)= X^3+X^2+1=[1\ 1\ 0\ 1]$

$$CRC = b(X) = \text{resto}\left[\frac{m(X)X^q}{g(X)}\right] = \text{resto}\left[\frac{(X^3+1)X^3}{X^3+X^2+1}\right] = \text{resto}\left[\frac{X^6+X^3}{X^3+X^2+1}\right]$$

$$= X+1$$

```
1 0 0 1 0 0 0      | 1 1 0 1
1 1 0 1            |---------
0 1 0 0 0            1 1 1 1
  1 1 0 1
  0 1 0 1 0
    1 1 0 1
    0 1 1 1 0
      1 1 0 1
      0 0 1 1
```

$$c(X) = m(X)X^3 + b(X) = (X^3+1)X^3 + (X+1).$$

$$= X^6 + X^3 + X + 1$$

$$= [1001 \quad 011]$$

CRC

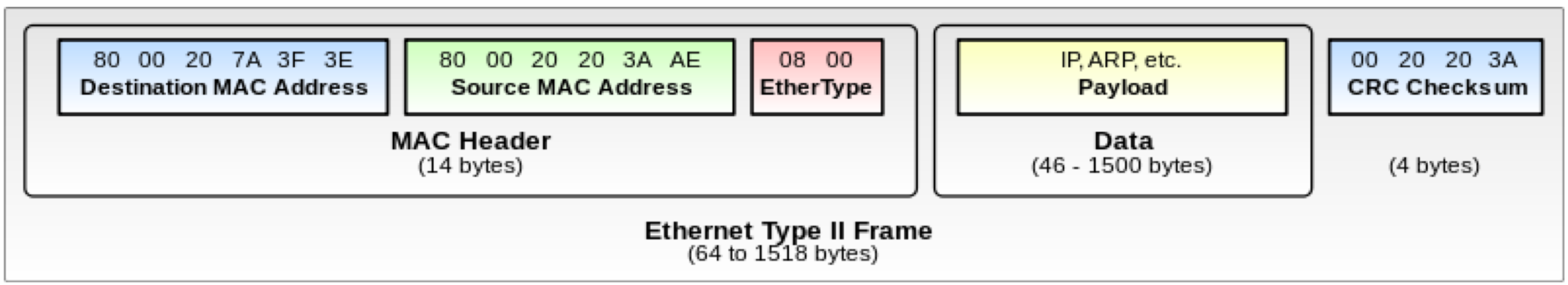
- Tipicamente é utilizado em modo de **detecção** de erros
- Quando a distância mínima do código for maior ou igual a 3, também pode ser usado em modo **correcção**
- Tipicamente temos um número reduzido de bits de paridade calculado para elevado número de bits de mensagem
  - n>>q>1
- O CRC tem elevada capacidade de detecção de erros, especialmente de *burst* de erros (rajada de erros)
- Um *burst* ou rajada de erros define-se como um bloco contíguo de bits recebidos em erro; o primeiro e último bit distam B bits entre si, sendo B o comprimento do *burst*

CRC

- Elevada capacidade de deteção de erros:
  - Todos os *burst* de dimensão q ou menor
  - Uma fração dos *burst* de dimensão q+1: $1 - 2^{-(q-1)}$
  - Uma fração dos *burst* de dimensão superior a q+1: $1 - 2^{-q}$
  - Todas as combinações de d_{min} ou menos erros
  - Todos os padrões com número ímpar de erros, quando o gerador tem número par de coeficientes não nulos
- Por exemplo, para o código CRC7 tem-se
  - Todos os burst de dimensão 7 ou menor
  - $1 - 2^{-(q-1)}$= $1 - 2^{-(7-1)} = 98{,}44\%$ dos burst de dimensão 8
  - $1 - 2^{-(q)}$= $1 - 2^{-(7)} = 99{,}2\%$ dos burst de dimensão superior a 8
  - Todos os padrões com número ímpar de erros

Aplicações

- Norma Ethernet 802.3 (LAN)
- Usa CRC32 (32bits) para verificação da integridade da trama
- O campo FCS – Frame Check Sequence no header da trama tem sempre 32 bits, independentemente da dimensão da trama



- Compressores, como o RAR, usam CRC32 para verificação da integridade de cada ficheiro comprimido
  - Antes de descomprimir o ficheiro, verifica a integridade do ficheiro

Bibliografia

- Meneses, C. “Comunicação de Dados”, Sebenta ADEETC - 2012
- Artur Ferreira, “Códigos detectors e corretores de erros”